RECEIVED
CENTRAL FAX CENTER
JUL 10 2007

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of: MACCHETTI ET AL.	Examiner: Carl G. COLIN
Serial No. 09/974,705	Art Unit: 2136
Filing Date: OCTOBER 10, 2001	Attorney Docket: 53537
For: METHOD AND CIRCUIT FOR DATA ENCRYPTION/DECRYPTION	

PRE-APPEAL BRIEF REQUEST FOR REVIEW

MS AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Responsive to the final Office Action of May 29, 2007, and in connection with the Notice of Appeal filed concurrently herewith, please consider the remarks set out below.

REMARKS

Applicants respectfully request reconsideration and that the current rejections of the Examiner be reversed.

I. The Claimed Invention

Independent Claim 31, for example, is directed to a device for converting data between an unencrypted format and an encrypted format. The device comprises a register for storing the data in the form of bit words, and a circuit for converting the data. The converting comprises performing a plurality of transformation rounds, with each transformation round comprising applying at least one transformation to a two-dimensional array of rows and columns of bit words defining a state array. The converting further comprises exchanging each of the rows with a respective column of the state array to form a transposed state array for at least one of the transformation rounds so that at least one transformation is applied to the transposed state array, and applying at least one round key to the state array in at least one of the transformation rounds. Independent Claim 21 is a method counterpart to Claim 31. Independent Claim 48 is

RECEIVED
CENTRAL FAX CENTER
JUL 10 2007

similar to Claim 21, but further recites using 8-bit words, and operating on a state array comprising a 4x4 matrix of 8-bit words.

II. The Claims Are Patentable

The Examiner rejected independent Claims 21, 31, and 48 over Ohkuma et al. in view of Luther. Ohkuma et al. discloses an apparatus for encrypting blocks of data. (Ohkuma et al.: Paragraphs 10-11). The encryption process occurs in multiple stages. (Paragraph 91-92). Ohkuma et al. also discloses that a matrix may be obtained by substituting rows, substituting columns, and arbitrarily transposing an arbitrary MDS matrix. (Paragraph 268). The Examiner correctly notes that Ohkuma et al. fails to disclose exchanging each of the rows with a respective column of the state array to form a transposed state array, as recited in independent Claims 21, 31, and 48. The Examiner looks to Luther to supply this deficiency.

Luther discloses an encryption system for two-dimensional data. The system of Luther encrypts through multiple encryption passes performed on binary data. In each pass, the mth row and the nth column of the binary data are encrypted. For each encryption pass, m and n are randomly selected and have a value, which is small relative to the size of the data. (Luther: Col. 1, lines 30-42).

The Examiner contends that steps S211 and S215 of Luther disclose exchanging each of the rows with a respective column of the state array to form a transposed state array, as in the claimed invention. Applicants submit that the Examiner is mischaracterizing Luther. Referencing the code of Luther depicted in Figure 8, reproduced below, and column 5, line 4 through column 6, line 18, in steps S201-202, the random generator is initialized. In step S203, the "StripeHeight" variable is set to a random value between two range values, for example, 1 and 5 (Figure 8).

In re Patent Application of
MACCHETTI ET AL.
Serial No. **09/974,705**
Filed: **OCTOBER 10, 2001**

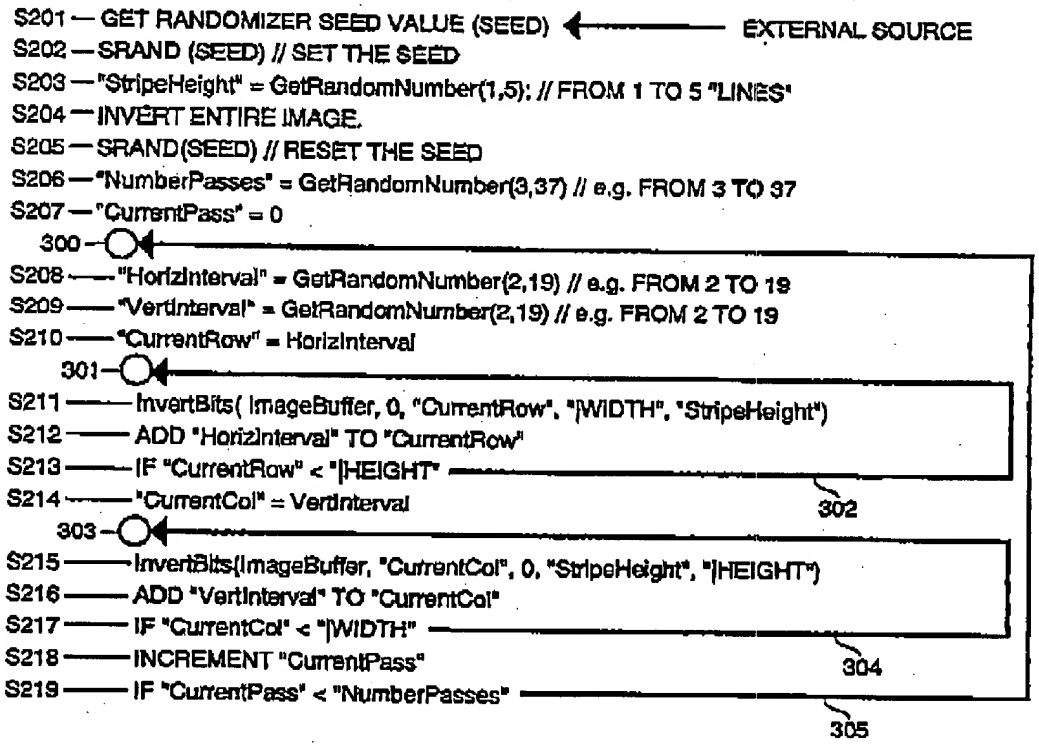

Figure 8 of Luther

In step S204, the entire image in the buffer is complemented. The random number generator is reset in step S205. In step S206, the number of encryption passes is randomly set; the "CurrentPass" variable is initialized to zero in step S207. Steps S208 and S209 randomly initialize the "HorizInterval" and "VertInterval" variables, respectively. The "CurrentRow" variable is randomly set in step S210. The invert-bit function (S211) is implemented iteratively.

On the first iteration, the invert-bit function is applied to a 2-dimesnional area of bits defined by the image width and the randomly set "StripeHeight" variable and located at the first column (0), and a row defined by the "CurrentRow" variable. The "CurrentRow" variable is then incremented by the random "HorizInterval" variable, then if the "CurrentRow" variable is

still less than the image height, i.e. not a nonexistent row. The process returns to step S211. In other words, the 2-dimensional area of bits being inverted is moved relatively vertically in the image array by a number of rows equal to the random "HorizInterval" variable. These steps (S211-213) are repeated until the "CurrentRow" variable is greater than or equal to the image height.

In steps S215-217, the invert-bit function is iteratively applied to a 2-dimensional area of bits defined by the image height and the "StripeHeight" variable. The position of application of the invert-bit function moves horizontally across the first row of the array, being iteratively applied to rows. The first column is set in step S214 to the random "VertInterval" variable. The next iteration is applied to a column being incremented by the "VertInterval" variable. This continues on until the "CurrentCol" variable exceeds the image width, i.e. nonexistent column. Thereafter, the "CurrentPass" variable is incremented, and if the appropriate number of passes has not been completed on the image array, the process restarts at step S208.

Applicants note that the invert-bit function does not transpose respective rows and columns, as claimed, but merely complements them. Moreover, for a proper transposition, the iterations of Luther being applied to the rows first and then columns would have to skip by equivalent intervals. As discussed above, steps S208 and S209 randomly initialize the "HorizInterval" and "VertInterval" variables, respectively. Moreover, Applicants note that Luther does not complement each row/column, but skips a number of rows/columns defined by the random "HorizInterval" and "VertInterval" variables, respectively. Accordingly, the row iterations and column iterations do not match up for a transposition but are instead random.

The Examiner's specifically contends that rows 3 and 4 are complemented and columns 4 and 5 are complemented, thereby disclosing the claimed transposition feature. Notwithstanding that complementing does not equal transposing, the Examiner contends that rows 3 and 4 are complemented in Figure 6, and that columns 4 and 5 are complemented in

Figure 7. Applicants note that this arrangement of Luther does not disclose the claimed transposition. Rather, the depicted iterations of Luther show complementing of rows 3-4, 6-7, 9-10, 12-13 (Figure 6: StripeHeight = 1, VertInterval =3 [inversion area = image width * (Stripeheight + 1)]) and show complementing of columns 4-5, 8-9, 12-13 (Figure 7: StripeHeight = 1, HorizInterval =4 [inversion area = (Stripeheight + 1)* image height]). This complementing of Luther is derived from the randomly generated "HorizInterval" and "VertInterval" variables.

Therefore, Applicants submit that Luther fails to disclose the claimed feature of exchanging each of the rows with a respective column of the state array to form a transposed state array. Accordingly, it is submitted that independent Claims 21, 31, and 48 are patentable over the prior art. Their respective dependent claims, which recite yet further distinguishing features, are also patentable over the prior art and require no further discussion herein.

Respectfully submitted,

JACK G. ABID
Reg. No. 58,237
Allen, Dyer, Doppelt, Milbrath
& Gilchrist, P.A.
255 S. Orange Avenue, Suite 1401
Orlando, Florida 32802
407-841-2330
Attorney for Applicants

CERTIFICATE OF FACSIMILE TRANSMISSION

I HEREBY CERTIFY that the foregoing correspondence has been forwarded via facsimile number 571-273-8300 to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 this 10th day of July, 2007.